

A Patient-Implantable Medical Records Storage Device

By: Michael Szczesniak

ECE-499, Computer Engineering Capstone 3

Advisor: Professor Spinelli

3/14/12

Report Summary

There are many flaws with the medical identification and records systems currently being used. Medical records are very often unorganized and scattered throughout many different organizations or institutions. Similarly, the methods currently being implemented to notify authorities of a serious medical condition are inefficient and outdated. The goal of this project is to address some of these issues through the development of an implantable device to store medical records.

This device will serve as a mobile storage device for your medical records, implanted within your body, so that your records stay with you wherever you go. This provides a much more organized method to store records as well as an easier way to retrieve necessary medical documents. It will also decrease the need for external notification bracelets, which could be lost in an accident. This will reduce the chance that a medical official isn't informed of a serious condition, which could result in harm to the patient.

The goal of this project is to present a proof of concept for this idea. Obviously it is not realistic to try and implant a device in someone for a senior project. However, the realistic goal is to design a working device, prove that it can be scaled down in size, and show that it can function inside a human body without presenting health risks. Despite many issues with both hardware and software, one way wireless communication was achieved. Future work could be done with the inductive powering, data encryption, and biocompatibility aspects of the device.

Table of Contents

Report Summary	2
Table of Figures and Tables	4
Introduction	5
Background	5
Design Requirements	8
Design Alternatives	10
Final Design and Implementation	12
Performance Estimates and Results	19
Production Schedule	22
Cost Analysis	24
User's Manual	26
Discussion, Conclusions, and Recommendations	27
References	30
Appendix A: Sample Code	31
Appendix B: Adapted Sample Code	39

Table of Figures and Tables

System Block Diagram	12
nRFgo Starter Kit Components	14
nRF24LU1+ Development Kit Components	15
Close-Up Image of RF Module and nRF24LU1+ Chip	16
Final System	19
Proposed Component Costs	24

Introduction

One area that has recently become of serious concern is the organization and tracking down of medical records. A person's medical records could be scattered all across the country, or even throughout the world, making it difficult to locate all of the necessary files in the event of an emergency. This report will discuss a project designed to present the feasibility of some parts of a patient-implantable medical records storage and retrieval system.

The goal of this project is to design an implantable medical device that will store an individual's medical records. In order to accomplish this task, the device needs to have three main subsystems: memory, powering, and data transmission. The device will be an inductively powered device with a memory chip for medical record storage. In the preliminary stages, this device will be geared towards the storage of the text-based portions of medical records, but could later be extended to deal with images as well.

Background

Historically, not much has been done to improve this problem. Hospitals keep track of their own patients' records and send them to other locations if need be. However, as addressed earlier, serious issues can arise in the event that records are not sent in time. The development and implementation of the device in this project could change society in many ways.

The first major way that a device like this could affect society is economically. In the past, whenever an institution needed medical records for a patient, they would have to request them from the location where they were being held. These files were then transported to the requested location. The request and transmission of these files costs money. Although the installation and implementation of this system would cost money initially, it could save money in the long run. Although time didn't allow for all subsystems of the device to be constructed, hardware that was purchased for the memory data transmission cost \$3.30 per chip, with an additional \$398 for development and starter kits for programming. For the remaining subsystem, although not explored in this project, a projected cost is in the region of \$110. In comparison to some of the other medical devices in a hospital, this total cost is a very minimal hit to any budget.

The next major aspect of this project is its great sustainability. This impressive sustainability is due to the fact that the device can be broken down into its 3 subsystems. Each of these subsystems has been implemented already, and with the growth of technology today, is constantly being improved. Therefore, the product can be easily maintained. Lastly, as the production of this system continues, the storage purposes of the device could be expanded to accommodate much larger data files such as images.

There are also a few ethical concerns with this project. The main concern is the protection of an individual's medical records. Medical records can sometimes contain extremely personal information that other individuals have no right to

unless provided permission. This will be dealt with by applying an encryption algorithm to the device. That way, no one can read from or write to the device unless they have permission and are using an appropriate reader with the necessary encryption key.

As with any implantable device, there are many health and safety concerns. The main focus with this device is to make sure that the power consumption and data transmission are both done so that there is no health risk. The data transmission will ideally be done using a frequency ranged specifically allocated by the Wireless Medical Telemetry Services. The power consumption will be modeled using currently produced implantable devices to ensure that it abides by safety standards.

One final aspect of this device that will affect society is the manufacturability of the system. As previously mentioned, this project is broken into three main subsystems. For each subsystem, there are many parts already in production, which proves that it is possible to manufacture the parts needed for the project. The way that these parts will tie together in the system is very similar to many devices in production. The main difference is that the entire device will be encapsulated in a biocompatible substance. This does not present any huge obstacle to overcome in order to manufacture the product. All in all, it will be very possible to manufacture such a product.

Design Requirements

The subsystems of this project will focus on powering the device, storing the data files, and transmitting the data. The main constraint of this project will be the size of the device, as it has to be implanted within a human. This size will need to be small enough such that it is easily implantable without irritation to the patient. Once implanted, the device will be very user-friendly. There will be no necessary maintenance or part replacements. Similarly, the use of the device will be extremely convenient, as it will allow a person to transfer a doctor their medical records in a matter of seconds. It will decrease the stress associated with being unsure if the records will be received in an appropriate time. The device will be very compatible in that it will transmit under a frequency range allocated specifically by the Wireless Medical Telemetry Services to avoid electromagnetic interference from other devices. Finally, there will be a large area for potential improvement. At the start of the project, a small amount of memory will be used to prove functionality. Once this is proved, the memory can be expanded to accommodate much larger files including x-rays and other medical images.

This device will serve one main purpose: centralizing all of a person's medical records in one chip that will be with the person at all times. The major components of the system will be a memory chip and a data reader. These two parts will communicate using a transmitter and receiver. The data reader will inductively power the device, providing power only when the device is being used. Inductively powering the device will accomplish two main goals. The first is that since power is

only provided when in use, it will reduce the power consumption of the device. Secondly, the use of inductive powering will eliminate the need to frequently replace the battery in the device. Also, since we are using inductive powering, the memory will be non-volatile so as not to lose any data stored in the device. Data will be protected via an encryption algorithm that can be selected in the later stages of development.

Ideally, the goal of transmission time is less than 15 seconds. Some files may be larger than others, but no one wants to stand next to a reader for an extended period of time. Since the device is dealing with text-based records, transmission in less than 15 seconds is very reasonable, even at a transmission rate of 1Mbps. In terms of reliability, the powering of the device must be very reliable. If it fails, a person would have to be cut open so that the device could be replaced. Slightly lower reliability for the data transmission is a little more acceptable, as this will only require the person to retransmit the data for a few more seconds. The cost of this project in the preliminary design stages is approximately \$600 for the entire system. Finally, human safety must remain paramount. A large factor in choosing parts will be selecting those that will reduce power consumption as much as possible without sacrificing functionality. Also, as previously mentioned, the device will transmit at a frequency set aside by the Wireless Medical Telemetry risks and recommendations, proving that it is a safe frequency at which to transmit through the human body.

Design Alternatives

A few alternatives were researched and discussed throughout the early stages of the project. The most controversial topic was the power subsystem of the device. The main possibilities that were taken into consideration include using a battery with a predetermined lifespan, using an inductively charged battery, and using an inductively powered battery. The latter of the three was chosen for the project.

The first option that was discussed for the powering of this device was a long-lasting battery. An example of such a battery exists in an EZ pass. A battery with a long lifespan was necessary for this project because it wouldn't be extremely easy to constantly replace the battery. However, regardless of how long the lifespan was, unless the battery lasted longer than the life of a person, it would have to be replaced. This presents a problem because people would be very reluctant to invest interest in this project if they had to be continuously cut open to replace the battery in the device.

The next powering system discussed was to use an inductively charged battery in the device. This was a better option than the aforementioned battery because you wouldn't have to cut someone open to replace it with a new battery. If using an inductively charged battery, you would simply have to put your body up to a "charger" so that your battery could replenish its power. However, there was a downside to this system since you would have to change your lifestyle to apportion

time to charging the device. This change in lifestyle could result in people not supporting the system.

Although this inductively charged battery did provide a step in the right direction, it still forced people to change their lifestyle to make the system work. From here it was decided that the battery would be inductively powered rather than inductively charged. This would eliminate the idea of charging the device so that people wouldn't have to go out of their way to charge it. Inductive powering was acceptable since data transmission didn't require too much power. Therefore, power could be provided to the device only when reading from or writing to the device.

Since it was decided that the device would be inductively powered, the memory would need to be non-volatile. The problem at hand was therefore determining what type of non-volatile memory to use such that power consumption can be reduced the most. The necessary components of the data transmission were very simple: a transmitter and a receiver pair, or a transceiver. Through product research, a microcontroller, produced by Nordic Semiconductor, was found that could actually combine these two remaining subsystems of memory and data transmission. These microcontrollers have flash memory and built in transceivers, both of which function at very low power consumption.

Final Design and Implementation

This project is broken up into three main subsystems: memory, data transmission, and powering. In the time provided for this project, key background research was done based on all three subsystems. However, given the time restrictions, part selection, purchase, and implementation was only achieved for the first two subsystems. A block diagram of the entire system can be seen in Figure 1.

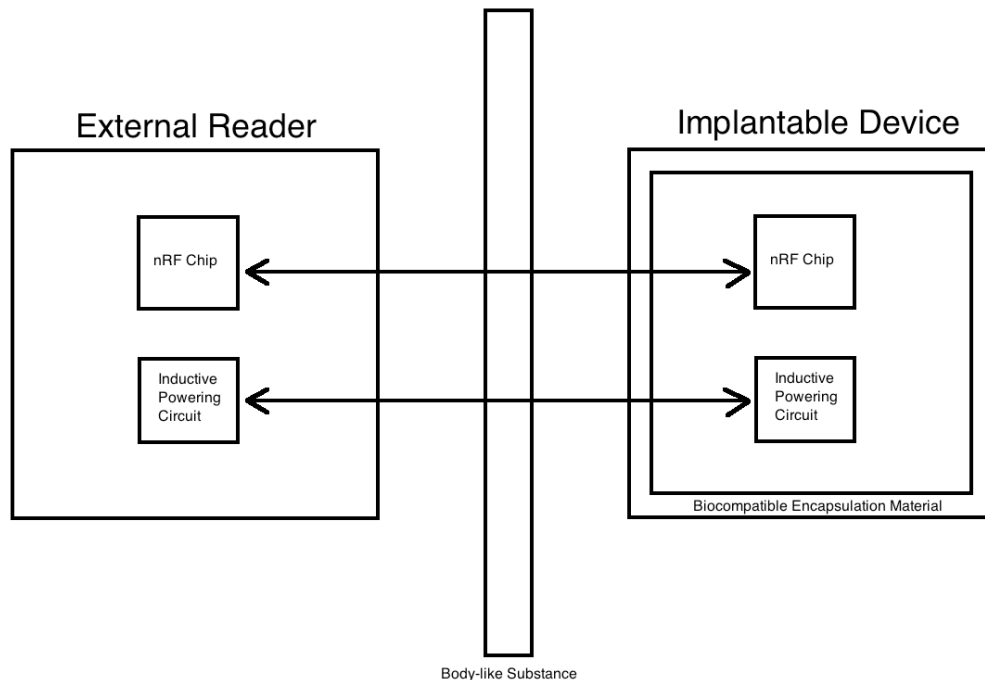


Figure 1: System Block Diagram

As seen in the image above, in a final production state, there would be a dividing section between the device and the data reader. This section will be composed of a skin-like substance, used to prove functionality upon implantation. The Data Reader will power the device inductively. The two will also communicate via the transceivers within the Nordic RF microcontrollers, to allow for doctors and other

medical officials to gain access to medical records when appropriate. This transmission will also allow for these officials to write to the device in the event of a new medical record being created. The device will be protected through an encryption algorithm. This will only grant access to the files when an authorized official is requesting access using the appropriate type of Data Reader. This encryption algorithm has not been selected yet, as there are many already in existence. As a result, focus was placed on the other more demanding aspects of the project. Finally, the implantable device will be encapsulated in some sort of biocompatible material such that, once implanted, the body does not reject it. Similar to the encryption algorithm, many biocompatible materials are already in production, so this was not a major focus given the time constraints.

A product manufactured by Nordic Semiconductor was chosen to accomplish the goals of both the memory storage and the data transmission subsystems. This part was a small chip called the nRF24LU1+. In order to work with this chip, a few other parts were needed as well, specifically the nRFgo Starter Kit and the nRF24LU1+ Development Kit. These parts, when used in conjunction with the nRF24LU1+ chip, formed a development system that was used to program the chip.

The first major part that was needed to work with the nRF24LU1+ chip was the nRFgo Starter Kit. This starter kit and its components can be seen in Figure 2.

The other main product that was needed to work with the nRF24LU1+ chip was the nRF24LU1+ Development Kit. The components of this development kit can be seen in Figure 3.



Figure 3: nRF24LU1+ Development Kit Components²

As seen in this figure, the development kit supplied two RF Modules and the necessary wires. It also came with five additional nRF24LU1+ chips, not pictured in the figure above, for testing. The RF modules also have the nRF24LU1+ chips mounted directly onto them. These modules can be planted directly on top of the nRFgo motherboards, creating the development system that was used for the programming of this project.

² <http://www.semiconductorstore.com/cart/pc/viewPrd.asp?idproduct=42896>

As previously noted, this development system is built around the selection of the nRF24LU1+ chip as the main component of the system. This chip was selected because it covers two of the main subsystems while still being extremely small. The actual size of this chip can be seen in a close-up image of the RF module, shown in Figure 4.

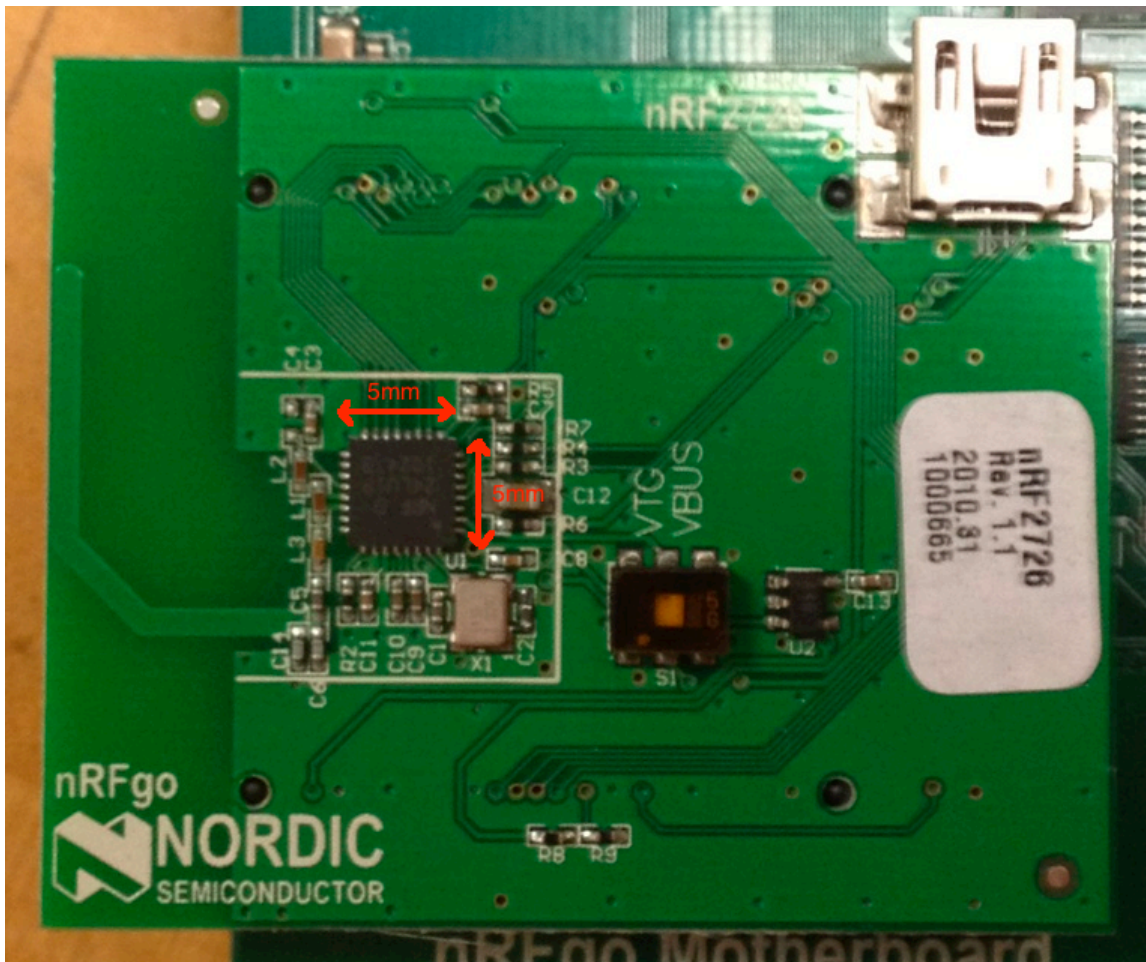


Figure 4: Close-Up Image of RF Module and nRF24LU1+ Chip

There are a few major components of this chip that make it so useful for this project. First of all, there is a 2.4GHz transceiver, which covers the data transmission subsystem of the project. The original goal was to work in a frequency range of 1395-1400 MHz or 1429-1432 MHz, which are the ranges allocated by the

Wireless Medical Telemetry risks and recommendations. These ranges are specifically set aside for medical telemetry devices such that electromagnetic interference can be avoided (US FDA). These frequency ranges apply well this project because according to work being done with very similar circuits at Ecole Polytechnique de Montreal, functionality is at its best in the GHz range (Sawan, Hu, and Coulombe). Although this original frequency range goal was not achieved in the current stage of the project, scaling down this frequency to approximately 1.4GHz in the later stages of production is a simple fix. The second major component of this chip is the flash memory, covering the data storage subsystem of the project. There is 16kB of flash memory located on this chip, which is more than enough for the text-based files in the preliminary stages of testing. This chip is also very small in size, totaling only 5mm by 5mm. It is also important to note that majority of the size most chips is due to the plastic casing, while the major components are much smaller. Therefore, this chip provided a great choice for the project as it could very feasibly be implanted in a person.

The final subsystem of this project was the inductive powering of the device. Unfortunately, due to the time constraints of the project, construction and implementation of this subsystem was not achieved. However, some design of a potential inductive circuit that could be used for this subsystem was done. This would be a very simple circuit, as we only needed to supply the device with approximately 3.3V. A few circuitry components were discussed to provide a more consistent supply voltage to the device so as not to burn out the chip or power it down during data transmission. Some of these components include zener diodes,

voltage regulators, and buck boost transformers. Given more time, further research and circuit construction could be achieved.

The overall goal of this project is to complete a proof of concept. It is not realistic to try and implant such a device in a person for a senior project. However, the subsystems were designed to be created and tested individually to confirm functionality. The implemented subsystems were then linked together to work as a single unit. In the later stages of production, once all subsystems are created, implemented, and tested, a skin-like barrier will be set in place to prove that it can function while transmitting through the substance. Lastly, it was necessary to choose small parts. This made it possible to prove that the size of the device could be scaled down to a realistically implantable size such that implantation wouldn't cause any irritation or discomfort.

The final system achieved in the time allotted for this project included two motherboards, two RF modules with mounted nRF24LU1+ chips, and a USB hub power supply. This system can be seen below in Figure 5.

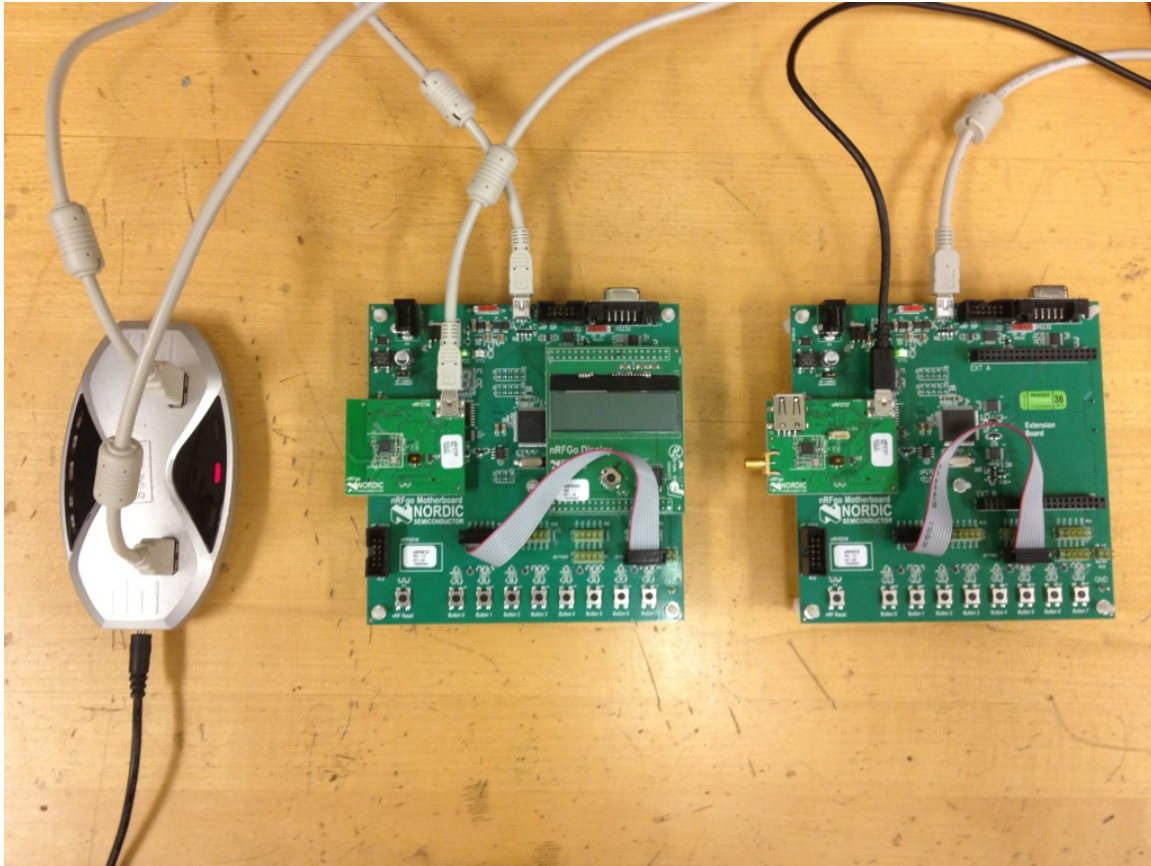


Figure 5: Final System

The hardware in this system allowed for some versatility in the choice of power supply. Either USB or AAA batteries could be used to power the motherboards. For the development system, it was decided to keep it simple during the preliminary stages and use the more consistent USB power. The USB hub in the picture allowed for powering of the system without the need for a computer with multiple USB ports.

Performance Estimates and Results

In the early design of this project, there were some preliminary estimated performance capabilities for the device. The main estimations involved

transmission time, power consumption, and ease of use. Thankfully, actual performance of the system ended up exceeding all of these estimations.

The first area of performance estimation was the data transmission subsystem. The original goal of the data transmission of the device was to achieve full transmission in less than 15 seconds. This seemed very reasonable, seeing as using a data rate of 1 Mbps would transmit 1MB of text records, a massive amount of data, in approximately 8 seconds. This was improved through the use of the Enhanced Shockburst capabilities of the nRF24LU1+ chip. This feature allowed for bi-directional data transmission, and by using the sample code provided by Nordic, a data rate of 2Mbps was achieved. This increased data rate would cut the transmission time of the proposed 1MB of data in half. Unfortunately, compiling issues prohibited further exploration of this feature, but it is very possible that this data rate could be improved even further.

The next area of performance estimation was power consumption of the device. The original design of the device included a transmitter, a receiver, and a flash memory chip. All of these devices would have their own individual power consumption. This consumption was minimized through the selection of using the nRF24LU1+ chip. This single chip contained all of the three parts mentioned in the original design. It also had a transceiver, eliminating the need for two separate parts for the data transmission, and thus reducing the power consumption even further. Overall, this chip minimized the power consumption of the device to a maximum of 180mW in the worst case.

The final area of performance estimation was ease of use for constructing and maintaining the device. As previously noted, there were three subsystems of this project: data storage, data transmission, and power supply. In the original design, the data storage subsystem had one component and the data transmission subsystem had two components. This would have required getting three separate components to function individually, and then to function in conjunction with one another. Through the choice of using the nRF24LU1+ chip, these three components were reduced to one, making the linking of subsystems much easier to deal with.

In the testing of the system constructed for this project, very quick data transmission was achieved. This testing was done using sample code, provided by Nordic Semiconductor, which explored the Gazell and Enhanced Shockburst features of the chip. This sample code can be seen in Appendix A. Gazell is a wireless connectivity stack solution used by Nordic. Enhanced Shockburst is a similar wireless transmission feature that allows for bi-directional data transmission through the use of acknowledgements and paired user data. This sample code paired the pushbuttons on one of the motherboards, deemed the “device”, with the LEDs on the other motherboard, deemed the “host.” As a pushbutton on the device was pressed, the corresponding LED on the host would light. The transmission completed by this code, however, was simply transferring one byte of data, specifically bits 0 through 5 since the nRF24LU1+ chips can only deal with 6 bits. Due to numerous compiling issues in Keil μ Vision 4, properly functioning compiled .hex files were not reachable. As a result of these tribulations,

this testing could not be expanded to deal with extended bi-directional data or larger data stored in the memory of the chip.

Production Schedule

The design of this project has taken a few turns of varying magnitude since the creation of the idea in the spring of 2011. In the original design, the device was to be powered inductively via some powering bracelet. In order to minimize the amount that this new technology would change a user's life, this was changed to have the data reader inductively power the device.

In the fall of 2011, the background research began for the project. Since the project was combining three separate technologies in a way that hadn't been commercially done yet, this background research took the majority of the fall trimester. After about 6 weeks of background research, it was time to begin searching for the appropriate parts for the project. This product search was done throughout weeks 7 and 8 of the fall term. Once reasonable parts were found, a proposal for a Student Research Grant was written and submitted in early November 2011. The remaining two weeks of fall term were spent writing and revising a "Design Report" – the final paper for ECE-498.

In the winter of 2011-12, the previously submitted grant proposal was returned with suggestions for resubmission. As a result, the first few weeks of winter term were spent continuing the product search and revising the proposal. This proposal was then resubmitted in late January 2012. With the product search

finally complete, and time winding down, parts were ordered at the same time as this resubmission.

Once the parts arrived, many hardware and software issues were experienced. The first major issue was the constant request for driver installations, which presented a huge problem due to restrictions on the student D4NEW accounts in Science and Engineering. Thankfully, Union College Lab Manager Gene Davison was very helpful in assisting with all of these tedious and consistent issues. These driver issues used up the majority of the time leading up to the final presentation and poster session. As a result, work had to be shifted to focus on completing the poster and PowerPoint. In order to get a working demonstration for the final presentation, with very little time left, focus was spent on getting Nordic Semiconductor's provided sample code to compile and function. This demonstration essentially showed the wireless communication and data transmission between the two RF modules. After the presentation, the remaining time in weeks 8 through 10 were spent trying to deal with many software issues. Unfortunately, due to time constraints, the focus in the remaining days of the term had to be steered towards finishing this final report and finalizing the project website.

The major flaw in the scheduling and planning of this project was that not enough time was allowed for potential hardware and software issues. In any project, you will run into errors that you did not foresee. These errors need to be accounted for in your scheduling. Given the chance to revise this project schedule,

part selection and purchase would have been done much earlier to allow more time for unforeseen errors that could come up.

Cost Analysis

In the early stages of this project, many hours were spent researching potential parts and their costs. The costs of the parts for which grant money was requested can be seen in Table 1.

Component	Source	Quantity	Cost
Nordic Semiconductor: NRF24LU1-F16Q32-T	SemiconductorStore.com	4	\$3.30
Nordic nRFgo Starter Kit: NRF6700	SemiconductorStore.com	1	\$399
Nordic nRF24LU1 Development Kit: NRF24LU1P-32-DK	SemiconductorStore.com	1	\$99
Circuit Board: DEV-00022	Sparkfun.com	1	\$30.95
Energizer Qi Inductive Charger: 1599396	BestBuy.com	1	\$89.99
Energizer Xi Inductive Charging Sleeve: 2088529	BestBuy.com	1	\$34.99

Table 1: Proposed Components Costs

The costs of some of the parts chosen early on in the design differed from those of the parts actually used. The biggest difference was in the transmitter and receiver pair. The majority of the parts found early on were systems where the transmitter and receiver were two separate parts. These parts also had their own development systems. Upon finding the Nordic Semiconductor nRF24LU1+ chip, we were able to accomplish the transmitter and receiver issue in one relatively cheap part, only \$3.30 per chip, which also covered our memory storage subsystem. The starter and

development kits did cost a bit more, but only maintenance technicians and users actually programming the chips would only require these. Use of this chip also minimized the need for a circuit board, furthering the minimization of the system cost.

The only subsystem for which parts weren't purchased was the inductive powering subsystem. Although the Energizer Qi system was proposed, this didn't fit well in our system for various reasons. Due to lack of appropriate products on the market that could fulfill the requirements of the system, it seems as if an inductive circuit would need to be produced specifically for this device. The cost of the components of this subsystem would be quite small, since circuit elements aren't very expensive. The production of these circuits would, however, entail a lot of work, which would drive the market price up.

There are a number of factors that go into the development of a market price for a device. Unfortunately, in the case of this project, a number of these factors are unknown thus far. Some of these factors include the cost of the biocompatible encasing chosen for the device, the cost of the construction of the data reader's body, and the projected cost accounting for the work put in to code development and circuit construction. These costs could be determined in a later stage of the project.

User's Manual

Since this project was a proof of concept and no device was actually produced, the following manual is a projected manual of an ideal manufactured device. Due to the simplicity in the design of the subsystems, their interaction, and the device as a whole, operation of the system will be extremely simple. Once the hospital or other office has installed their data reader, and the patients have their device implanted, there are very few steps to be taken. First, you must confirm that the data reader is powered on. Once this is confirmed, the administrator will then choose for the reader to either read records from the device, or write a new selected record to the device. The patient can then place their arm up the reader. This data reader will then inductively power the device and complete the function chosen by the administrator. The patients arm must be held up to the reader until notified by the reader that the function has been completed successfully.

Maintenance of the device, due to the simplicity of the design and the system components, is also very simple. Since the device is inductively powered, it will not need to be removed from the body to exchange any batteries. The functions carried out by this system are also very reusable. As a result, once the code is written for these functions, it is very unlikely that they will need further debugging. In the event of continuous failed transmission, there are a few things that can be checked. Since the implanted device is encapsulated in a protective biocompatible material, it is very unlikely that the device itself is damaged. It is also very unlikely that the implanted chip was fried, due to the fact that the inductive circuit has built in

components to regulate the voltage and restrict overpowering or frying the device. If an error occurs, the first thing to check is whether or not voltage is being transferred between the two devices. If no voltage is transferred, there is probably an issue with the inductive circuit within the reader. If voltage is being transferred, it is likely that the chip in the reader experienced some issues. In either case, the issue is most likely due to overuse and can be solved by simply replacing the malfunctioning part of the data reader. It is also very important to note that the reader is used on a daily basis, while the implanted device is not. Therefore, the hardware in the reader is much more prone to some sort of issue, while the chances of experiencing such an issue with the implanted device remains incalculably lower.

Discussion, Conclusions, and Recommendations

Disorganization of medical records presents a huge concern in today's society. Failure to acquire a patient's records in an emergency could cause serious harm to the patient, and in some cases, death. This project worked to present the possibility of a device that could allow hospitals to avoid such a serious issue. This is an implantable device that works to store and protect text-based medical records, as well as wirelessly transmit the data to and from an external data reader.

Throughout the project, a number of serious issues were experienced, both hardware and software related. The hardware issues all had to do with driver installations. Unfortunately, working on the computers in the lab and not having administrator access, these problems were very tedious and time consuming since

only the administrator could solve them. Once all of these issues were taken care of, I could move on to dealing with the code of the project.

The main approach taken with this project was to essentially observe and test the sample code provided by Nordic Semiconductor, and then to move on to try and expand this code in a manner that would suit the project goals. This was first exercised using the Gazell sample code. After observing the functionality of the code via the two-motherboard development system, a variation of the code was attempted. This variation combined the “host” and “device” code to attempt bi-directional data transmission. After compiling successfully, testing with the development system revealed that the code didn’t function. I then moved on to dealing with the Enhanced Shockburst sample code. Again, testing with the system yielded positive functionality. Unfortunately, the sample code for this was quite basic and didn’t include any use of acknowledgements. It was then brought to my attention by my advisor that the compiled .hex file might not be working even if the code was functional. To test this, I created a new project in μ Vision 4 and added the Enhanced Shockburst files. The compiled file surprisingly did not work when tested with the system, even though it was the exact sample code from Nordic that I had already tested. This made it very clear that a number of software issues were being experienced. Many actions were taken to try and fix this problem including downloading new and updated versions of all software being used, changing different options in the project in μ Vision 4, and including different files to the projects. None of these yielded positive results, and unfortunately, time had run out.

Due to the multitude of issues, this project has been a great learning experience. The first major lesson learned was that when scheduling and planning your project, you always have to allow more time than expected for debugging unforeseen errors. No matter how well you predict what problems you may encounter, there are always going to be unexpected setbacks. Unfortunately in the case of this project, those setbacks were massively time consuming in a situation that didn't allow for many problems. The second major lesson learned was that you should always do plenty of background research on the software that will be used in your project. Not only should you learn what programs you will need to use, but you should also begin familiarizing yourself with these programs and reach out for help from people who have used them before. Unfortunately, the documentation of the products purchased for this project was very minimal, so it wasn't very clear what compiler would be used. As a result, not much work could be done ahead of time to try and avoid the compiling issues experienced in this project. The final major lesson learned was that when purchasing any product that comes with software, always check that the software is up to date. Products very often sit on the shelves while the software in the box goes through many versions. By simply updating the software used in this project, many mindboggling issues were solved.

All of the sample code mentioned in this paper can be seen in Appendix A, while the experimental and adapted sample code can be seen in Appendix B.

References

- "Guidance for Industry - Wireless Medical Telemetry Risks and Recommendations." *U S Food and Drug Administration Home Page*. Web. 22 Nov. 2011.
<<http://www.fda.gov/MedicalDevices/DeviceRegulationandGuidance/GuidanceDocuments/ucm070920.htm>>.
- "NRF24LU1 - 2.4GHz RF - Products - Nordic Semiconductor." *Ultra Low Power Wireless Solutions from NORDIC SEMICONDUCTOR*. Web. 14 Mar. 2012. <[http://www.nordicsemi.com/eng/Products/2.4GHz-RF/nRF24LU1P/\(language\)/eng-GB](http://www.nordicsemi.com/eng/Products/2.4GHz-RF/nRF24LU1P/(language)/eng-GB)>.
- "NRF24LU1 Development Kit - 2.4GHz RF - Products - Nordic Semiconductor." *Ultra Low Power Wireless Solutions from NORDIC SEMICONDUCTOR*. Web. 14 Mar. 2012.
<<http://www.nordicsemi.com/eng/Products/2.4GHz-RF/nRF24LU1-Development-Kit>>.
- "NRFgo Starter Kit - 2.4GHz RF - Products - Nordic Semiconductor." *Ultra Low Power Wireless Solutions from NORDIC SEMICONDUCTOR*. Web. 14 Mar. 2012. <<http://www.nordicsemi.com/eng/Products/2.4GHz-RF/nRFgo-Starter-Kit>>.
- Sawan, Mohamad, Yamu Hu, and Jonathan Coulombe. "Wirelessly Powered and Bidirectional Data Exchanged in Smart Medical Microsystems." *IEEE 2005 Custom Integrated Circuits Conference* (2005): 5-12. *IEEE*. Web.

Appendix A: Nordic Sample Code

```
/* Copyright (c) 2009 Nordic Semiconductor. All Rights Reserved.
 *
 * The information contained herein is confidential property of Nordic
 * Semiconductor ASA. Terms and conditions of usage are described in detail
 * in NORDIC SEMICONDUCTOR STANDARD SOFTWARE LICENSE AGREEMENT.
 *
 * Licensees are granted free, non-transferable use of the information. NO
 * WARRANTY of ANY KIND is provided. This heading must NOT be removed from
 * the file.
 *
 * $LastChangedRevision: 133 $
 */

/** @file
 * @brief Gazell Link Layer Host example
 * @defgroup gzll_host_example Gazell Link Layer Host example
 * @{
 * @ingroup nrf_examples
 *
 * @brief This example listens for data and writes the first byte (byte 0) of the
 * received payloads to P0.
 *
 * Protocol parameters such as addresses and channels are specified
 * in @ref gazell_examples/params/gzll_params.h.
 *
 * The project @ref gzll_device_example can be used as a counterpart for transmitting the
 * data.
 */

#include "gzll_mcu.h"
#include "gzll.h"

void main(void)
{
    uint8_t payload[GZLL_MAX_PAYLOAD_LENGTH];

    mcu_init();
    gzll_init();

    // Set P0 as output
    P0DIR = 0;

    EA = 1;

    // Enter host mode (start monitoring for data)
    gzll_rx_start();
```

```

for(;;)
{
    // If data received
    if(gzll_rx_fifo_read(payload, NULL, NULL))
    {
        // Write received payload[0] to port 0
        P0 = payload[0];
    }
}
}
/** @} */

```

```

/* Copyright (c) 2009 Nordic Semiconductor. All Rights Reserved.

```

```

*

```

```

* The information contained herein is confidential property of Nordic
* Semiconductor ASA. Terms and conditions of usage are described in detail
* in NORDIC SEMICONDUCTOR STANDARD SOFTWARE LICENSE AGREEMENT.
*

```

```

* Licensees are granted free, non-transferable use of the information. NO
* WARRANTY of ANY KIND is provided. This heading must NOT be removed from
* the file.
*

```

```

* $LastChangedRevision: 133 $

```

```

*/

```

```

/* Copyright (c) 2009 Nordic Semiconductor. All Rights Reserved.

```

```

*

```

```

* The information contained herein is confidential property of Nordic
* Semiconductor ASA. Terms and conditions of usage are described in detail
* in NORDIC SEMICONDUCTOR STANDARD SOFTWARE LICENSE AGREEMENT.
*

```

```

* Licensees are granted free, non-transferable use of the information. NO
* WARRANTY of ANY KIND is provided. This heading must NOT be removed from
* the file.
*

```

```

* $LastChangedRevision: 133 $

```

```

*/

```

```

/** @file

```

```

* @brief Gazell Link Layer Device example
* @defgroup gzll_device_example Gazell Link Layer Device example
* @{
* @ingroup nrf_examples
*

```

```

* @brief This example sends packets continuously. The contents of P0 are
* sent in the first payload byte (byte 0).
*

```

```

*

```

```

*

```

```

* Protocol parameters such as addresses and channels are specified

```

```

* in @ref gazell_examples/params/gzll_params.h.
*
* The project @ref gzll_host_example can be used as a counterpart for receiving the data.
*
*/

//lint -e534

#include "gzll_mcu.h"
#include "gzll.h"

void main(void)
{
    uint8_t payload[GZLL_MAX_PAYLOAD_LENGTH];

    mcu_init();
    gzll_init();

    EA = 1;

    for(;;)
    {
        // If gazell link layer idle
        if(gzll_get_state() == GZLL_IDLE)
        {
            // Put P0 contents in payload[0]
            payload[0] = ~P0;
            // Transmits payload[] to pipe 0 address
            gzll_tx_data(payload, GZLL_MAX_FW_PAYLOAD_LENGTH, 0);
        }
    }
}
/** @} */

/** @file
 * @brief Enhanced ShockBurst Primary Receiver example
 * @defgroup esb_prx_example Enhanced ShockBurst Primary Receiver (PRX) example
 * @{
 * @ingroup nrf_examples
 *
 * @brief This example monitors for data and writes the first byte (byte 0) of the
 * received payloads to P0.
 *
 * The example shows the minimum required setup for receiving packets from a
 * primary transmitter (PTX) device.
 *
 * The following default radio parameters are being used:
 * - RF channel 2
 * - 2 Mbps data rate
 * - RX address 0xE7E7E7E7E7 (pipe 0) and 0xC2C2C2C2C2 (pipe 1)

```

```

* - 1 byte CRC
*
* The project @ref esb_ptx_example can be used as a counterpart for transmitting the data.
*
*/

//lint -e717
//lint -e534
//lint -e714
//lint -e783

#ifdef MCU_NRF24LE1
#include "nrf24le1.h"
#endif

#ifdef MCU_NRF24LU1P
#include "nrf24lu1p.h"
#endif

#include <stdint.h>
#include "hal_nrf.h"

// Global variables
uint8_t payload[3];

void main()
{
    #ifdef MCU_NRF24LU1P
    // Enable radio SPI
    RFCTL = 0x10;
    #endif

    // Set P0 as output
    P0DIR = 0;

    // Enable the radio clock
    RFCKEN = 1;

    // Enable RF interrupt
    RF = 1;
    // Enable global interrupt
    EA = 1;

    // Configure radio as primary receiver (PTX)
    hal_nrf_set_operation_mode(HAL_NRF_PRX);

    // Set payload width to 3 bytes
    hal_nrf_set_rx_payload_width((int)HAL_NRF_PIPE0, 3);

    // Power up radio

```

```

hal_nrf_set_power_mode(HAL_NRF_PWR_UP);

// Enable receiver
CE_HIGH();

for(;;){
}

// Radio interrupt
NRF_ISR()
{
    uint8_t irq_flags;

    // Read and clear IRQ flags from radio
    irq_flags = hal_nrf_get_clear_irq_flags();

    // If data received
    if((irq_flags & (1<<(uint8_t)HAL_NRF_RX_DR)) > 0)
    {
        // Read payload
        while(!hal_nrf_rx_fifo_empty())
        {
            hal_nrf_read_rx_payload(payload);
        }

        // Write received payload[0] to port 0
        P0 = payload[0];
    }
}
/** @} */

/* Copyright (c) 2009 Nordic Semiconductor. All Rights Reserved.
*
* The information contained herein is property of Nordic Semiconductor ASA.
* Terms and conditions of usage are described in detail in NORDIC
* SEMICONDUCTOR STANDARD SOFTWARE LICENSE AGREEMENT.
*
* Licensees are granted free, non-transferable use of the information. NO
* WARRANTY of ANY KIND is provided. This heading must NOT be removed from
* the file.
*
* $LastChangedRevision: 230 $
*/

/** @file
* @brief Enhanced ShockBurst Primary Transmitter example
* @defgroup esb_ptx_example Enhanced ShockBurst Primary Transmitter (PTX) example
* @{
* @ingroup nrf_examples
*

```

```

* @brief This example sends packets continuously. The contents of P0 are
* sent in the first payload byte (byte 0).
*
* The example shows the minimum required setup for transmitting packets to a
* primary receiver (PRX) device.
*
* The following default radio parameters are being used:
* - RF channel 2
* - 2 Mbps data rate
* - TX address 0xE7E7E7E7E7
* - 1 byte CRC
*
* The project @ref esb_prx_example can be used as a counterpart for receiving the data.
*
*/

```

```

//lint -e717
//lint -e714
//lint -e640

```

```

#ifdef MCU_NRF24LE1
#include "nrf24le1.h"
#endif

```

```

#ifdef MCU_NRF24LU1P
#include "nrf24lu1p.h"
#endif

```

```

#include <stdint.h>
#include <stdbool.h>
#include "hal_nrf.h"
#include "hal_nrf_reg.h"
#ifdef MCU_NRF24LU1P
#include "hal_clk.h" //lint !e322 !e7 "include file not found"
#endif

```

```

// Global variables
bool radio_busy;

```

```

void main()
{
    uint8_t payload[3];

```

```

#ifdef MCU_NRF24LU1P
// Enable radio SPI
RFCTL = 0x10;
#endif

```

```

// Enable the radio clock
RFCKEN = 1;

```

```

// Enable RF interrupt
RF = 1;

// Enable global interrupt
EA = 1;

// Wait for the xtal to power up
#ifndef MCU_NRF24LU1P
while(hal_clk_get_16m_source() != HAL_CLK_XOSC16M) //lint !e40 "Undeclared identifier"
{
}
#endif
// Power up radio
hal_nrf_set_power_mode(HAL_NRF_PWR_UP);

for(;;)
{
    // Put P0 contents in payload[0]
    payload[0] = ~P0;

    // Write payload to radio TX FIFO
    hal_nrf_write_tx_payload(payload, 3);

    // Toggle radio CE signal to start transmission
    CE_PULSE();

    radio_busy = true;

    // Wait for radio operation to finish
    while((radio_busy) > 0){
    }
}

// Radio interrupt
NRF_ISR()
{
    uint8_t irq_flags;

    // Read and clear IRQ flags from radio
    irq_flags = hal_nrf_get_clear_irq_flags();

    switch(irq_flags)
    {
        // Transmission success
        case (1 << (uint8_t)HAL_NRF_TX_DS):
            radio_busy = false;
            // Data has been sent
            break;
    }
}

```

```

// Transmission failed (maximum re-transmits)
case (1 << (uint8_t)HAL_NRF_MAX_RT):
    // When a MAX_RT interrupt occurs the TX payload will not be removed from the TX
    FIFO.
    // If the packet is to be discarded this must be done manually by flushing the TX FIFO.
    // Alternatively, CE_PULSE() can be called re-starting transmission of the payload.
    // (Will only be possible after the radio irq flags are cleared)
    hal_nrf_flush_tx();
    radio_busy = false;
    break;
default:
    ;
}
}
/** @} */

```

Appendix B: Adapted Sample Code

```
/* Copyright (c) 2009 Nordic Semiconductor. All Rights Reserved.
*
* The information contained herein is confidential property of Nordic
* Semiconductor ASA. Terms and conditions of usage are described in detail
* in NORDIC SEMICONDUCTOR STANDARD SOFTWARE LICENSE AGREEMENT.
*
* Licensees are granted free, non-transferable use of the information. NO
* WARRANTY of ANY KIND is provided. This heading must NOT be removed from
* the file.
*
* $LastChangedRevision: 133 $
*/

/** @file
* @brief Gazell Link Layer 2-way Communication example
* @defgroup gzll_2wayComm_example Gazell Link Layer Device example
* @{
* @ingroup nrf_examples
*
* @brief This example sends packets continuously. The contents of P0 are
* sent in the first payload byte (byte 0). It listens for data and writes
* the first byte (byte 0) of the received payloads to P0.2 and P0.3 of motherboard.
* Bits P0.0 and P0.1 are attached to buttons for input. Bits P0.2 and P0.3 are
* wired to LED's 0 and 1 for coordinated output.
*
* Protocol parameters such as addresses and channels are specified
* in @ref gazell_examples/params/gzll_params.h.
*
*/

//lint -e534

#include "gzll_mcu.h"
#include "gzll.h"

void main(void)
{
    uint8_t payload[GZLL_MAX_PAYLOAD_LENGTH];
    uint8_t temp;

    mcu_init();
    gzll_init();

    EA = 1;

    // Configure P0 for bits 0 and 1 as input, 2 and 3 as output
```

```

P0DIR = 0xFC;

for(;;)
{
//HOST PART OF CODE
//initialize in receiver mode
    gzll_rx_start();

    // If data received
do
{
    // Write received payload[0] to port 1
    P0 = payload[0];
} while(gzll_rx_fifo_read(payload, NULL, NULL));

//DEVICE PART OF CODE
    //change to idle mode
    //gzll_goto_idle();
    // If gazell link layer idle
    if(gzll_get_state() == GZLL_IDLE)
    {
        // Put P0 contents in temp and shift left by 2 (P0.2 and P0.3 are attached to
        // LEDs 0 and 1). Then put in payload[0].
        temp = ~P0;
        temp = temp<<2;
        //temp = temp & 0xFC; //not necessary, simply ensures that
the two least significant bits are 0
        payload[0] = temp;
        // Transmits payload[] to pipe 0 address
        gzll_tx_data(payload, GZLL_MAX_FW_PAYLOAD_LENGTH, 0);
    }
}
}

/** @} */

```